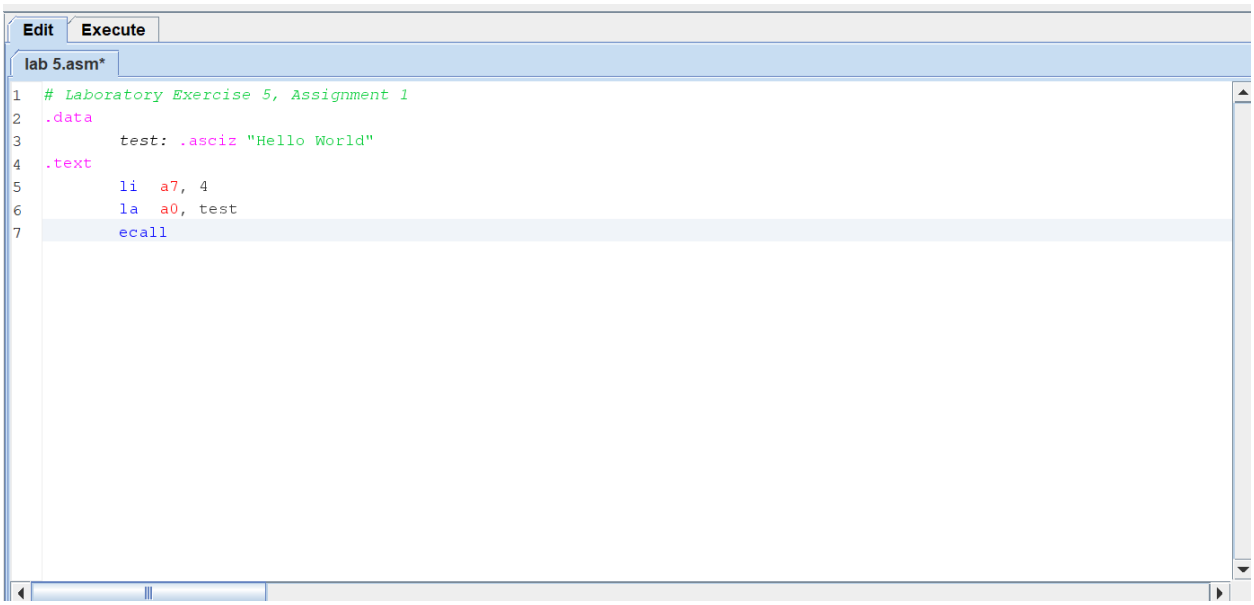
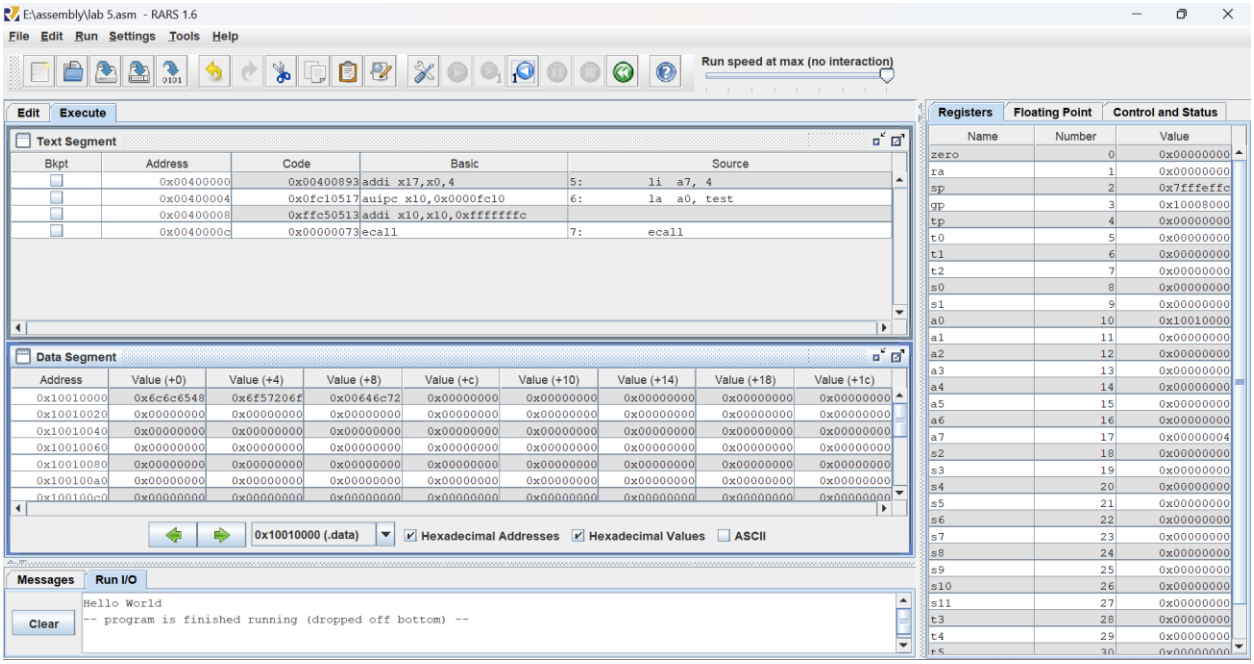
Student name: Lê Ngọc Anh Vũ

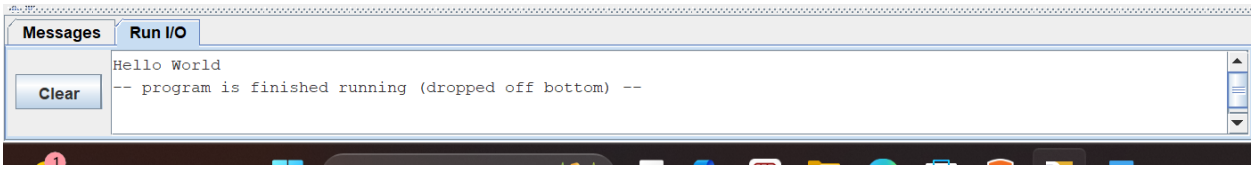
Student ID: 20236014

Assignment 1: Create a new project to implement the program in Home Assignment 1. Compile and upload to simulator. Run and observe the result. Go to Data Segment, check how test string is stored and packed in memory.



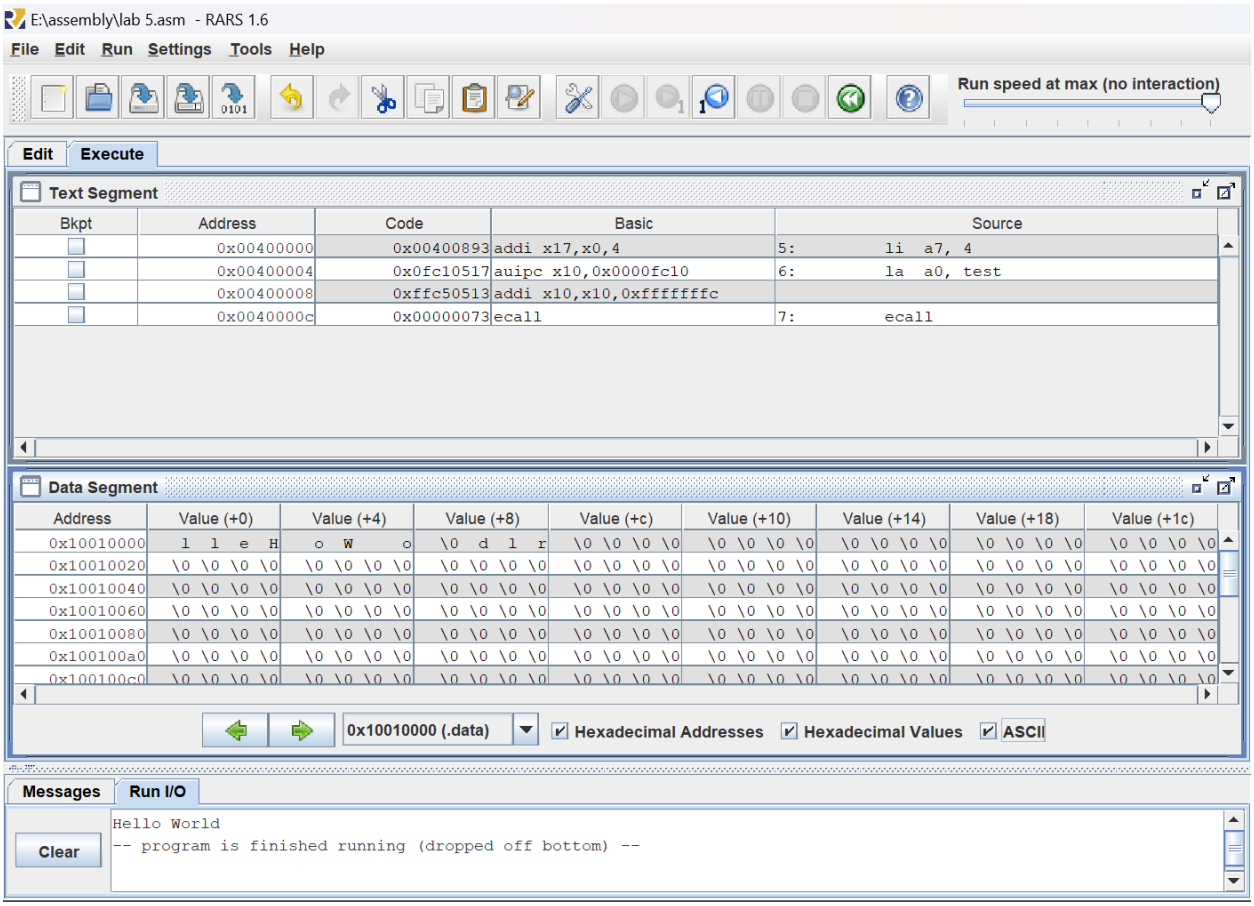


After running the program, the Run I/O represents the test string “Hello World” as the picture below:



Display data segment values in ASCII.

Test string is stored and packed as the picture below:



Assignment 2: Create a new project to print the sum of two register $s0 and $s1 according to this format:

"The sum of (s0) and (s1) is (result)"

The source code is as below:

We assign the register s0 to 4, the register s1 to 1

We is going to print :" The sum of 4 and 1 is 5"

Edit Execute

lab 5.asm*

```
1  # Laboratory Exercise 5, Assignment 2
2  .data
3          mess1: .asciz "The sum of "
4          mess2: .asciz " and "
5          mess3: .asciz " is "
6  .text
7          addi s0, zero, 4
8          addi s1, zero, 1
9
10         li a7, 4
11         la a0, mess1
12         ecall
13
14         li a7, 1
15         add a0, s0, zero
16         ecall
17
18         li a7, 4
19         la a0, mess2
20         ecall
21
22         li a7, 1
```

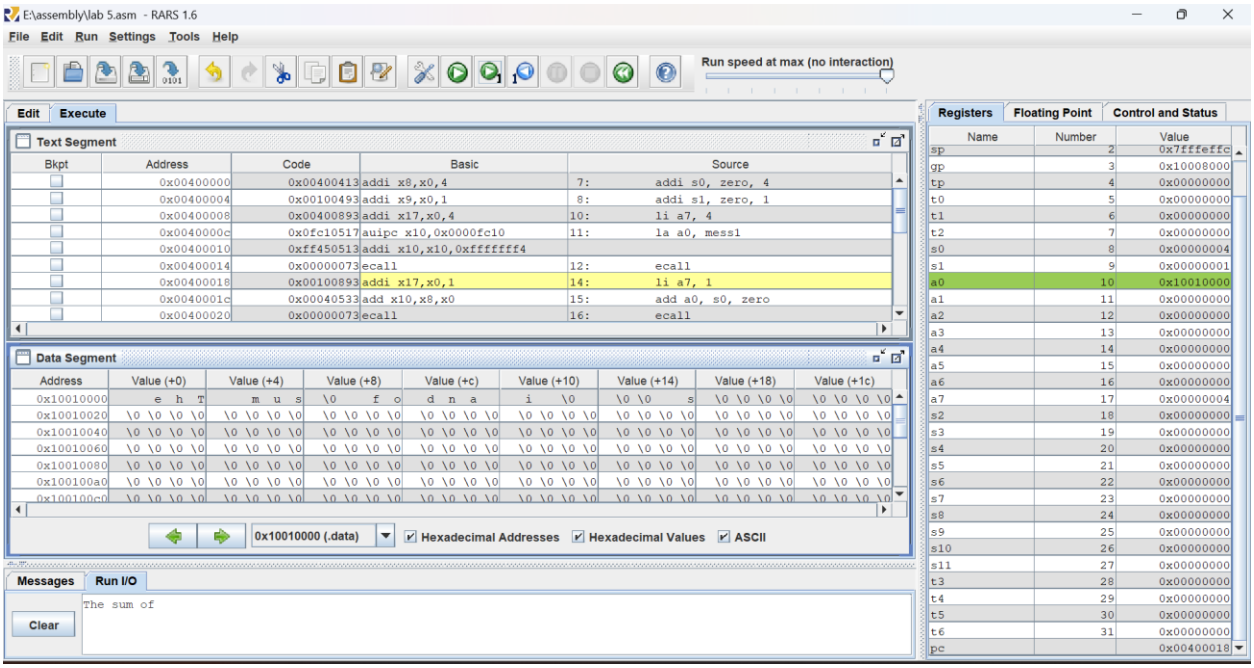
Edit Execute

lab 5.asm*

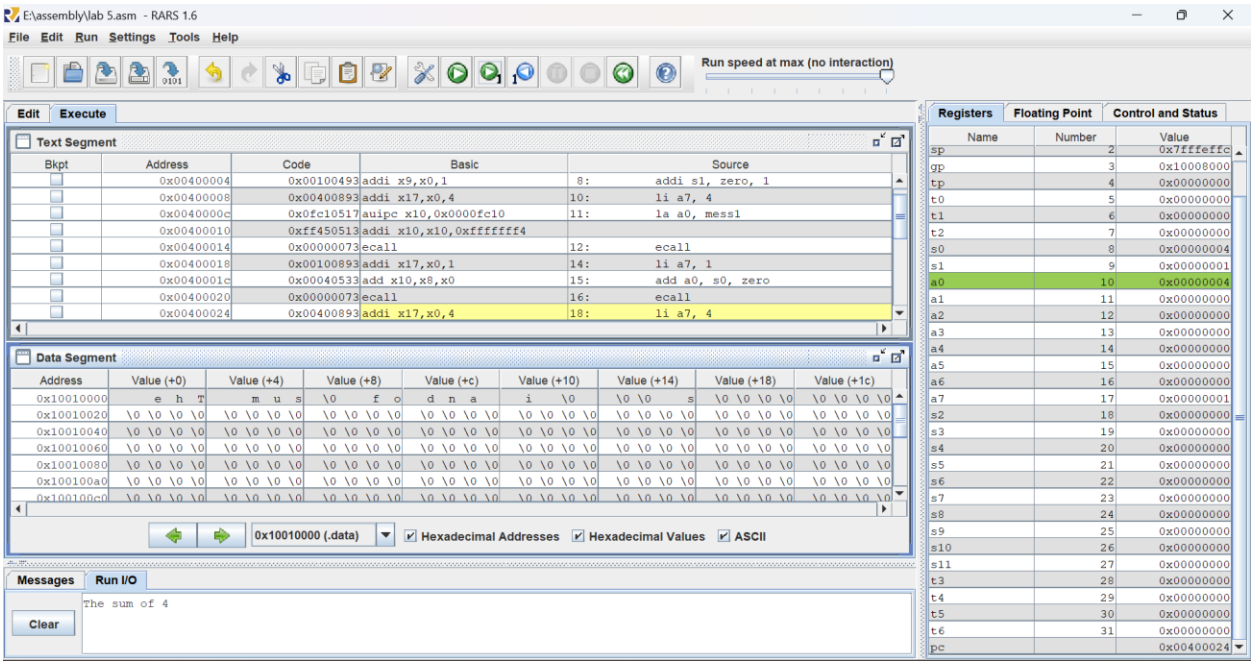
```
12         ecall
13
14         li a7, 1
15         add a0, s0, zero
16         ecall
17
18         li a7, 4
19         la a0, mess2
20         ecall
21
22         li a7, 1
23         add a0, s1, zero
24         ecall
25
26         li a7, 4
27         la a0, mess3
28         ecall
29
30         li a7, 1
31         add a0, s0, s1
32         ecall
```

Line: 8 Column: 18 Show Line Numbers

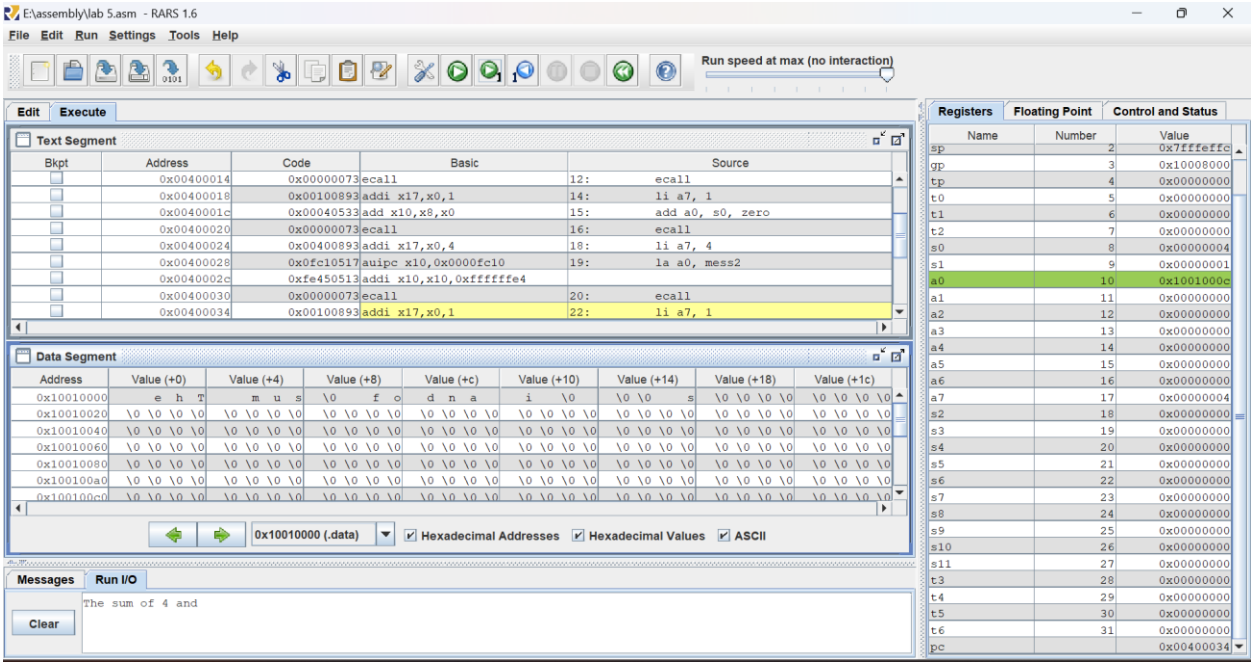
The first ecall



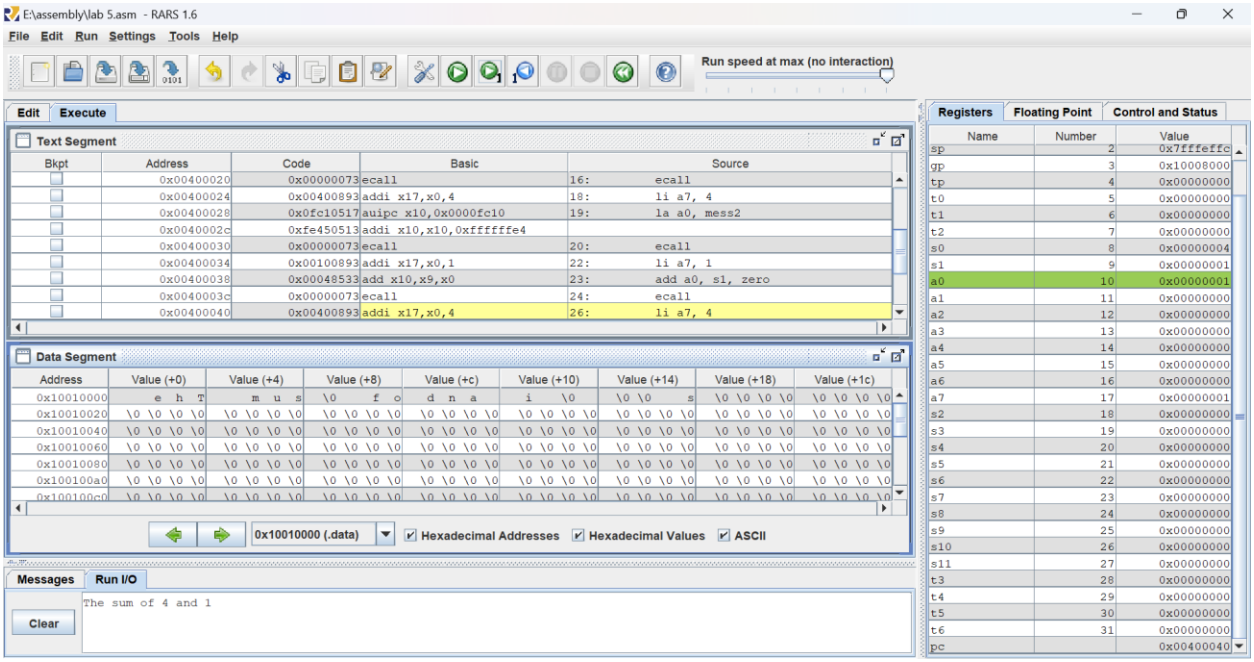
The second ecall



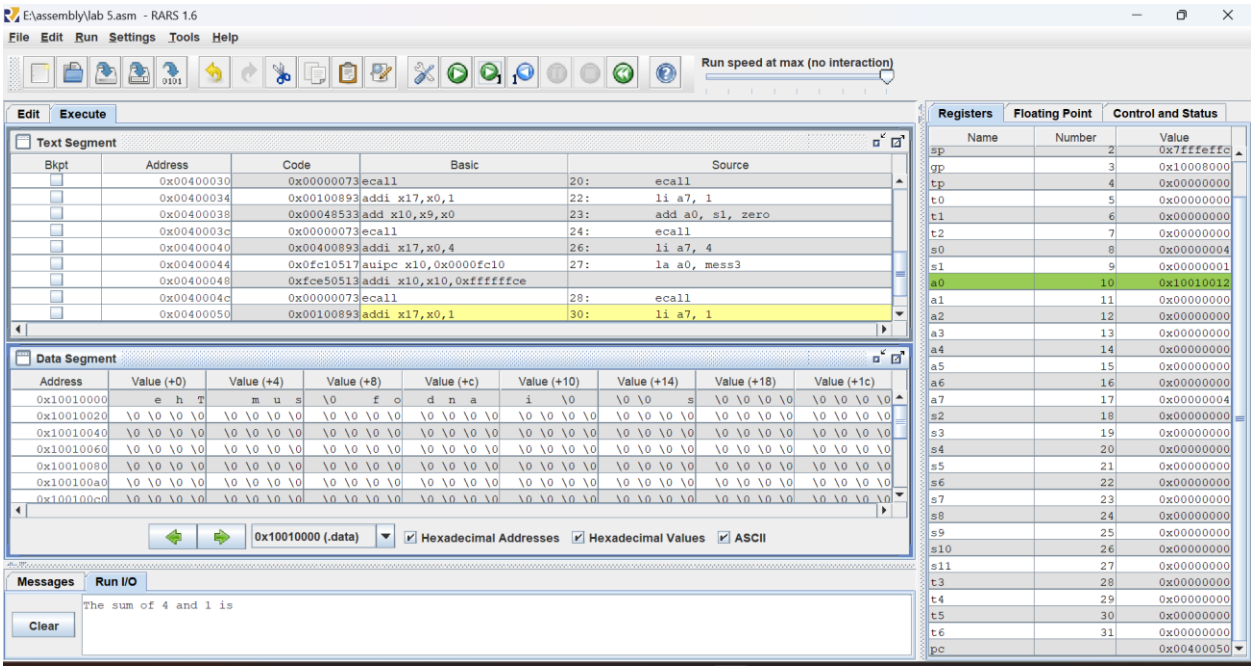
The third ecall



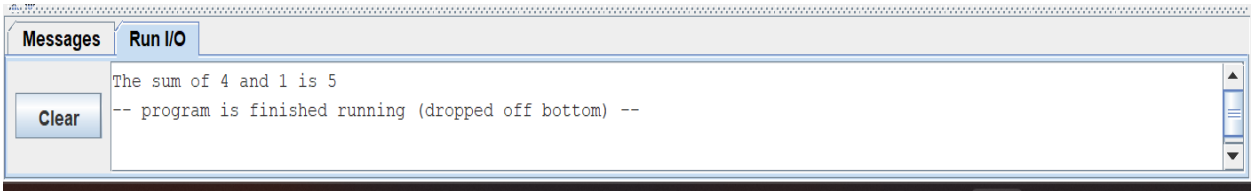
The fourth ecall

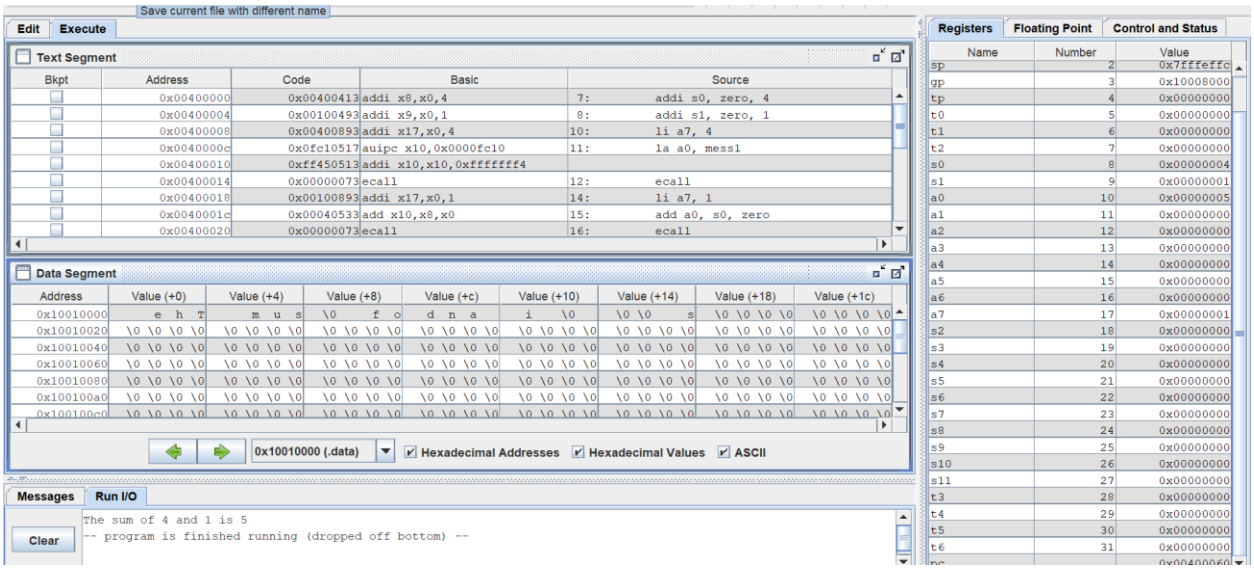


The fifth ecall



Finally, after the last ecall, the Run I/O print “The sum of 4 and 1 is 5” correctly as our plan:

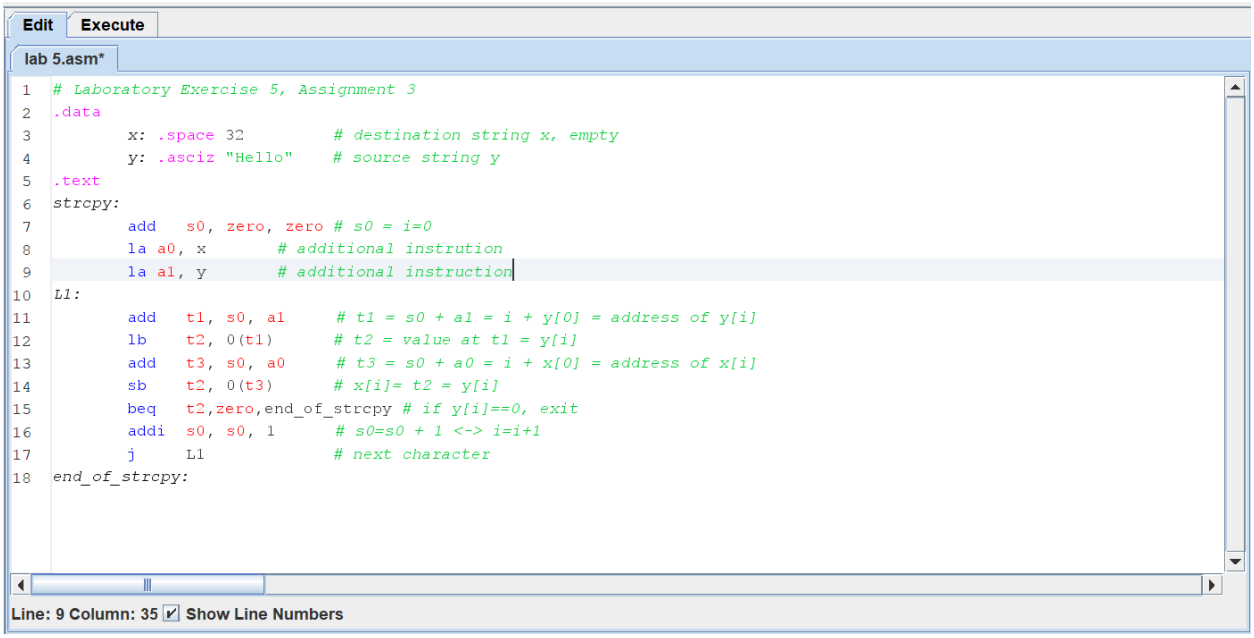




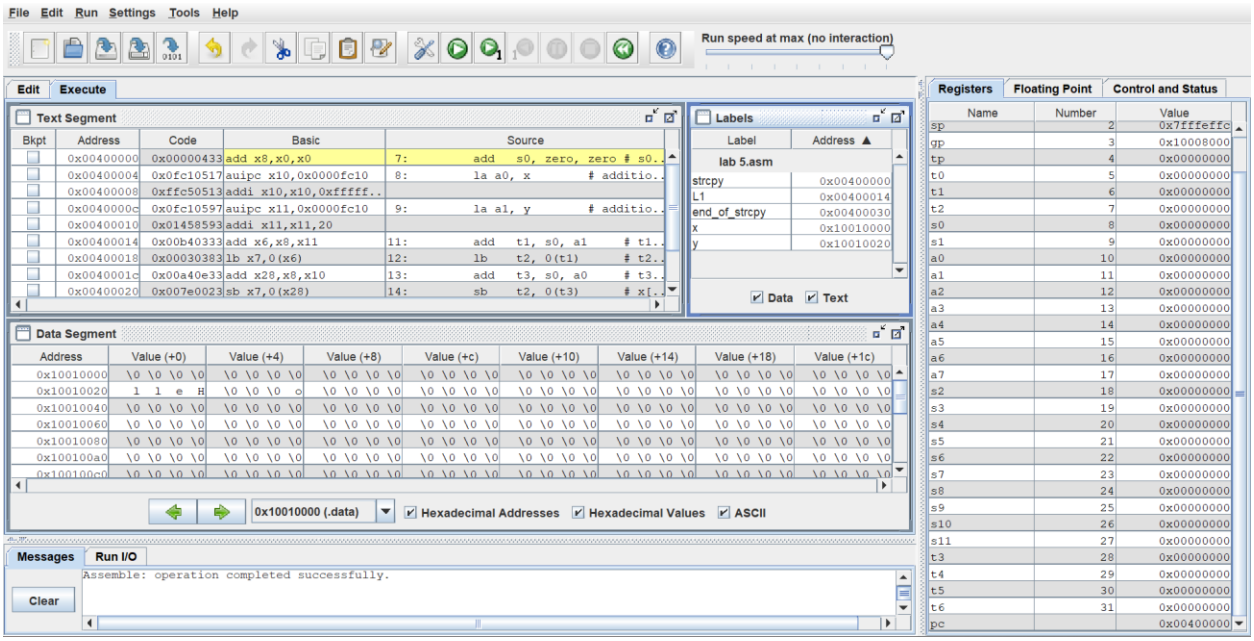
Assignment 3: Create a new project to implement the program in Home Assignment 2. Add more instructions to assign a test string for y variable, and implement strcpy function. Compile and upload to simulator. Run and observe the result

Here is the source code with two additional instructions:

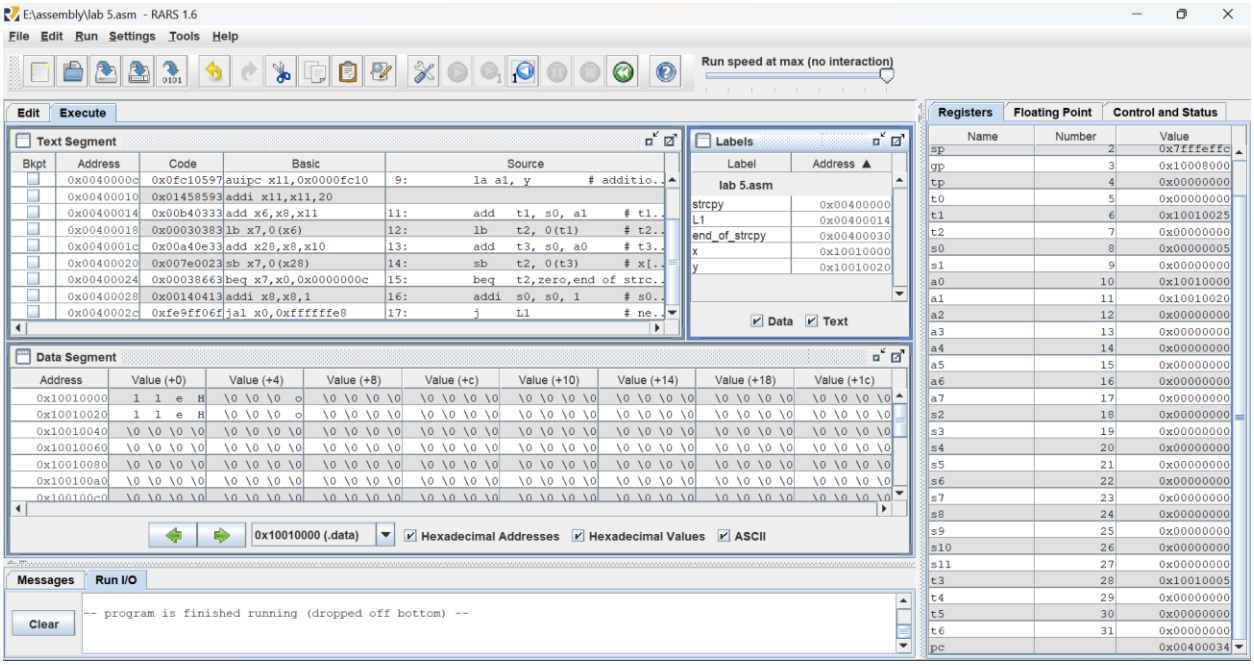
We load the address of label x and y into the registers a0 and a1 respectively.



Before running the program, in Data Segment, we can see that the address of label y is 0x10010020 and y stores the string “Hello”. The address of label x is 0x10010000 and x does not store any thing

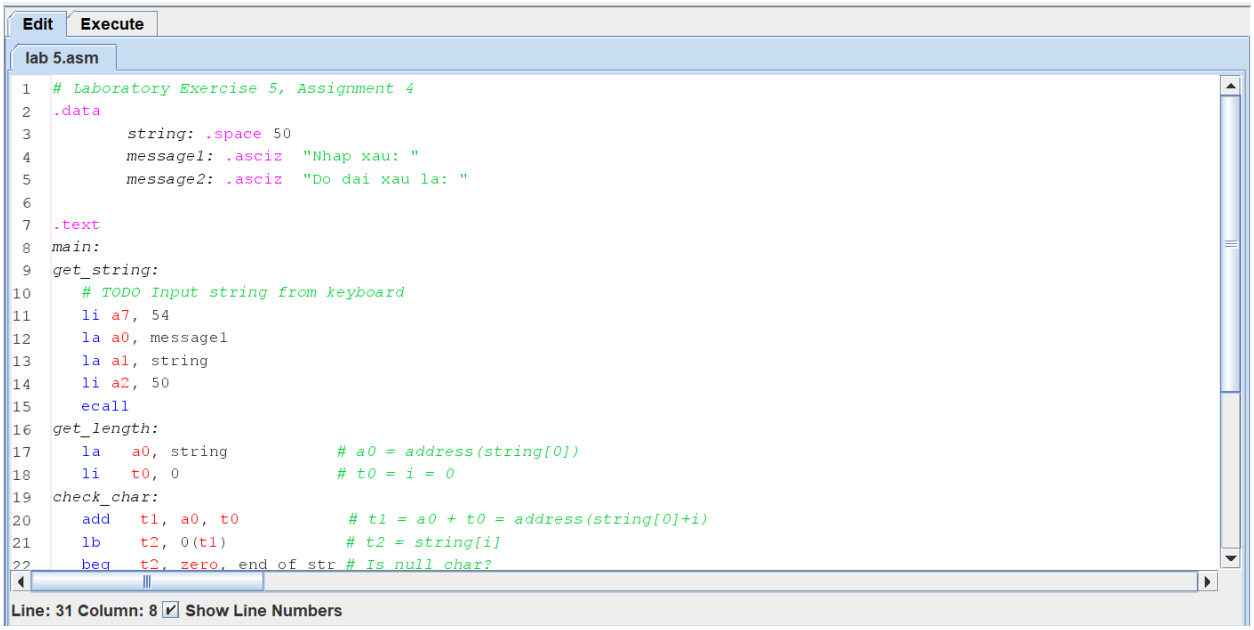


After running the program, we can see string “Hello” in the address of lael x. It means that x has already copied the string from y.



Assignment 4: Accomplish the Home Assignment 3 with ecall function to get a string from dialog, and show the length to message dialog.

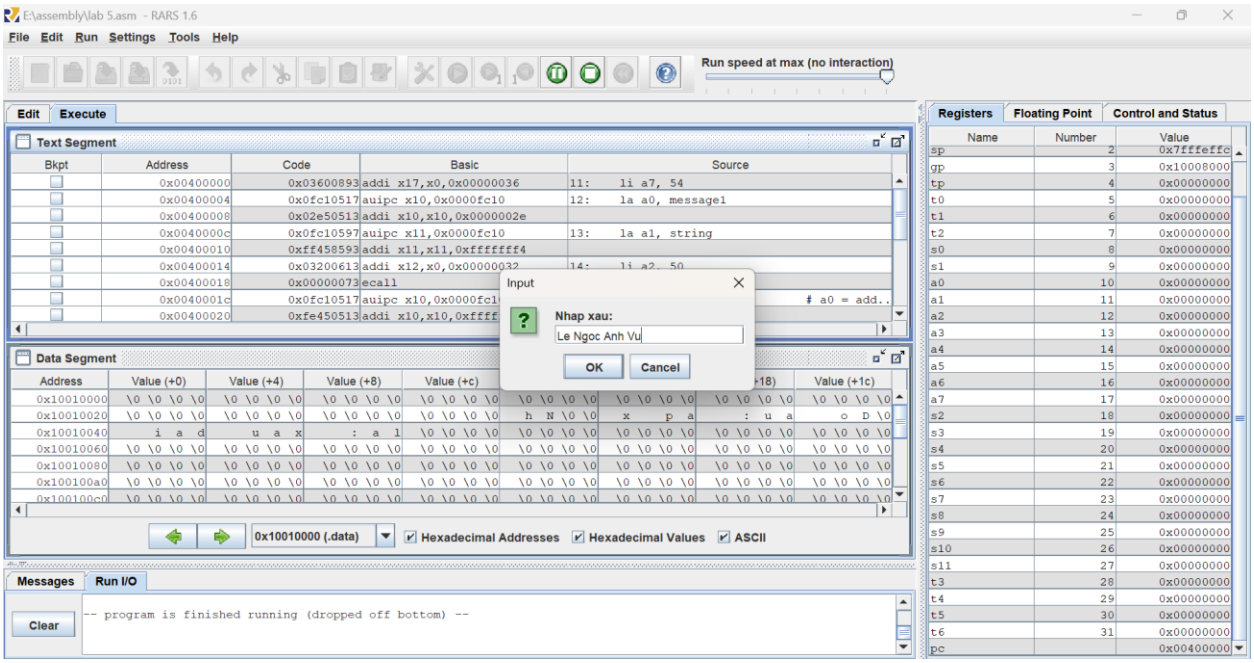
My source code is as below:



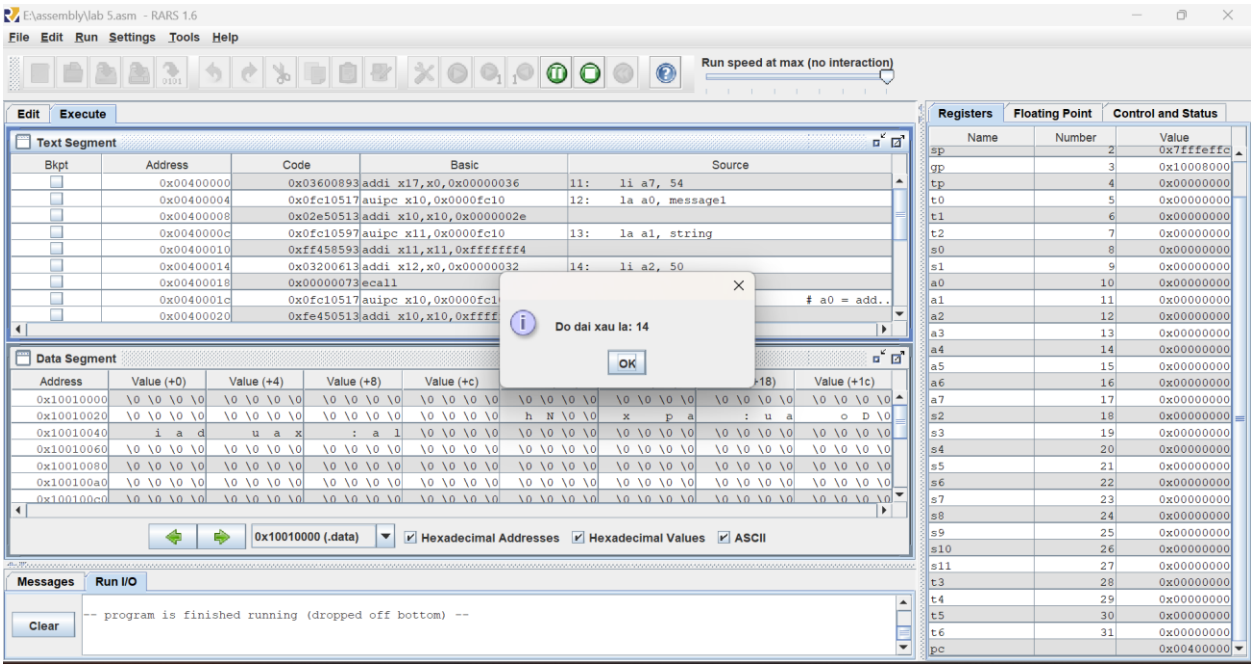
```
12    la a0, message1
13    la a1, string
14    li a2, 50
15    ecall
16  get_length:
17    la   a0, string           # a0 = address(string[0])
18    li   t0, 0                # t0 = i = 0
19  check_char:
20    add   t1, a0, t0           # t1 = a0 + t0 = address(string[0]+i)
21    lb    t2, 0(t1)            # t2 = string[i]
22    beq   t2, zero, end_of_str # Is null char?
23    addi  t0, t0, 1            # t0 = t0 + 1 -> i = i + 1
24    j  check_char
25  end_of_str:
26  end_of_get_length:
27  print_length:
28    # TODO print result to console
29    li a7, 56
30    la a0, message2
31    addi a1, t0, -1
32    ecall
```

Line: 31 Column: 8 Show Line Numbers

We enter a string:



And then, the dialog sends the message about the length of the string.



Assignment 5: Write a program that let user input a string. Input process will be terminated when user press Enter or then length of the string exceed 20 characters. Print the reverse string.

My source code is as below:

lab 5.asm

```
1  # Laboratory Exercise 5, Assignment 5
2  .data
3          string: .space 21
4          message1: .asciz  "Enter a string: "
5          message2: .asciz  "The reverse string: "
6
7  .text
8  main:
9          li t1, 10
10         li t0, 20
11         li s0, 0        # count the number of characters
12         li a7, 4
13         la a0, message1
14         ecall
15 get_string:
16    li a7, 12
17    ecall
18    beq a0, t1, end
19    addi s0, s0, 1
20    sb a0, 0(sp)
21    addi sp, sp, -1
22    beq s0, t0, end1
```

Line: 26 Column: 14 Show Line Numbers

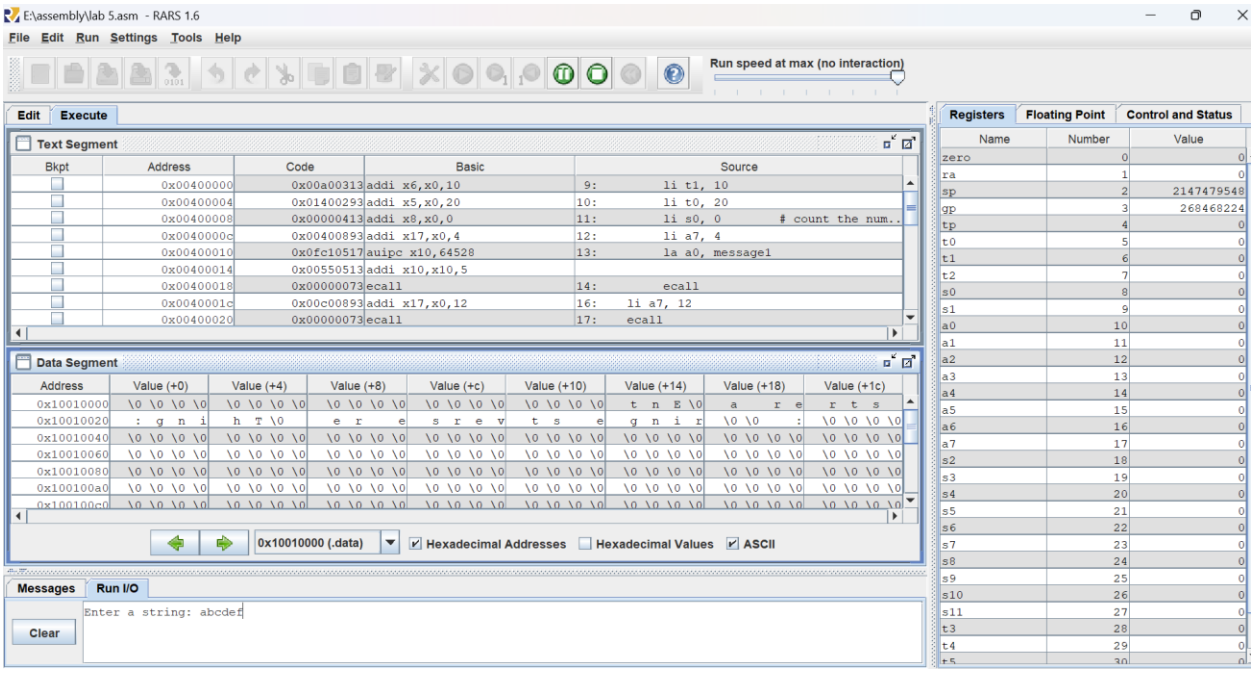
Edit Execute

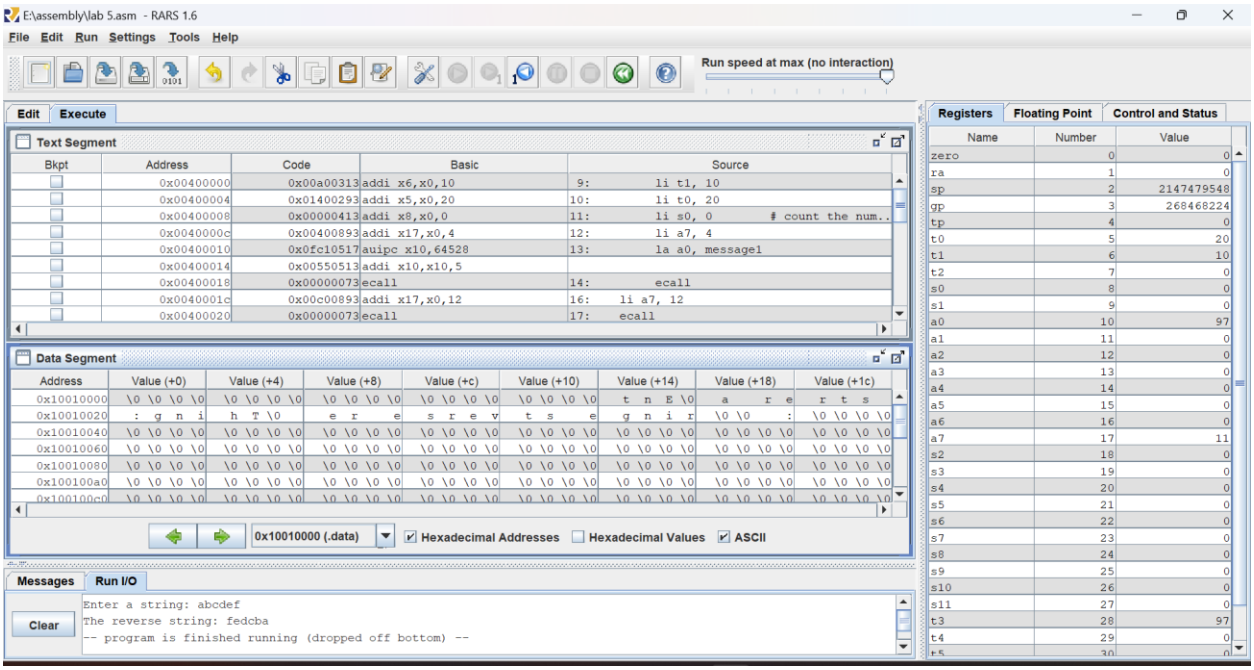
lab 5.asm

```
21    addi sp, sp, -1
22    beq s0, t0, end1
23    j get_string
24 end1:
25    li a7, 11
26    li a0, '\n'
27    ecall
28 end:
29    li a7, 4
30    la a0, message2
31    ecall
32 reverse:
33    addi sp, sp, 1
34    lb t3, 0(sp)
35    addi s0, s0, -1
36    li a7, 11
37    add a0, t3, zero
38    ecall
39    beqz s0, finish
40    j reverse
41 finish:
```

Line: 26 Column: 14 Show Line Numbers

Case 1: the process input is going to terminated when I press Enter and the program will print the reverse string.





Case 2: the process input is going to terminated when the length of string exceeds 20 characters.